Biosystems Engineering (2005) **90** (4), 419–431
doi:10.1016/j.biosystemseng.2004.12.005
PH—Postharvest Technology

Performance of a System for Apple Surface Defect Identification in Near-infrared Images

B.S. Bennedsen; D.L. Peterson

USDA, Agricultural Research Service, Appalachian Fruit Research Station, Kearneysville, WV 25430, USA;
e-mail of corresponding author: bsb@kvl.dk

(Received 15 May 2004; accepted in revised form 9 December 2004; published online 12 February 2005)

This paper reports the development and testing of machine vision systems for sorting apples for surface defects, including bruises. The system operated on apples, which were oriented with the stem/calyx axis perpendicular to the imaging camera. Grey-scale images in the visible wavebands were used to verify orientation. Images for detection of defects were acquired through two optical filters at 740 and 950 nm, respectively. Defects were detected using a combination of three different threshold segmentation routines and one routine based on artificial neural networks and principal components. The paper reports quantitative measurement of the performance of the system for verification of orientation and a combination of the four segmentation routines. The routines were evaluated using eight different apple varieties. The ability of the routines to find individual defects and measure the area ranged from 77 to 91% for the number of defects detected, and from 78 to 92·7% of the total defective area.
Published by Elsevier Ltd on behalf of Silsoe Research Institute

1. Introduction

Surface defects are of great concern to producers of apples, as the perceived quality is highly related to the appearance of the fruit. Sorting apples for surface defect is mainly done manually, even though sorting equipment has been developed and is commercially available. However, many producers, especially in the USA, feel that the existing machinery is not sufficiently selective, especially on darker coloured varieties like Red Delicious.

The system reported here was built at the USDA Appalachian Fruit Research Station, Kearneysville, West Virginia, in close collaboration with Cornell University. The development of the sorter started in 1980, reaching its present configuration as described by Throop *et al.* (1999, 2003a). The design of the sorter distinguished itself by incorporating an orienting system, which aimed at orienting the apples with the stem–calyx axis perpendicular to the image-capturing camera and hence out of sight of the camera. This design eliminates the need for distinguishing between defects and stem/calyx, which is a critical part of the development of automatic apple sorters. Several researchers have tried to address this problem with varying success, *e.g.* Crowe and Delwiche (1996) and Leemans *et al.* (2000) which achieved error rates between 9 and 33%.

The sorting system, which is intended as a prototype for testing and developing the concept, consists of a conveyor in which the apples are transported in individual cups. There is a small wheel at the bottom of the cup, which turns the apples until the cavity at the stem or calyx is over the wheel and it thus looses contact with the apple surface. This will leave the apple with the stem/calyx axis vertical (Throop *et al.*, 2003b). After the orienting section, the apples enter the image acquisition area. This consists of a lighting system and a camera. When passing this area, the apples are tilted by 45°. This makes them rest against rollers, which turn the apples through one turn as the apples travel the length of the area, which is about 50 cm.

Experiments with the orienting system showed that orientation was successful for more than 95% of the apples, depending on variety (Throop *et al.*, 2003b).

With up to 5% of the apples incorrectly oriented there was a risk that these would be wrongly classified as defective and hence lost. Bennedsen *et al.* (2004) described the development and initial testing of the orientation verification system. In a simulated test with images of apples acquired under laboratory conditions and images classified by unsupervised feature extraction, correct identification of non-oriented apples occurred in about 97% of the cases.

With the system in its current configuration, detecting defects in the stem/calyx area is not possible. However, according to the concept as described by Throop *et al.* (2000), orienting the apples allows for the use of additional cameras, attuned to inspect the stem and calyx region. Hence, with more cameras, three sets of images could be acquired for each apple: one covering the stem region, another the calyx region, and a number of images covering the circumference as the apple rotates in front of the camera. Additional research and development is needed in order to add stem and calyx inspection to the system.

The image-capturing system was based on research by Throop and Aneshansley (1997). They investigated the performance of different wavelengths for detecting surface defects, and found that 540 nm produced the best segmentation of defects caused by the blister spot, early frost damage, powdery mildew, russet and sunburn. Seven hundred and fifty nano meters performed best for bitter pit, Botryosphaeria rot, chemical damage, Codling moth, corking, cracking, fly speck, hail damage, leaf roller, rot, scab and sooty blotch. The optimal wavelength for detecting bruises, punctures, and scald was 950 nm. Aneshansley *et al.* (2003) developed an optical filter system with a splitter and two band pass filters. The optics in the splitter divided the incoming radiation into three identical parts, each directed to a sub-area on the image sensor, thus enabling the camera to capture three images simultaneously.

Segmentation routines were developed for identifying defects in the near-infrared (NIR) images, captured by the imaging system. Two approaches were followed: one was based on artificial neural networks, and the other used a combination of thresholding methods.

The current paper presents the final testing of the combined system of orientation verification and image-processing routines on eight different apple varieties.

1.1. *Objectives*

The objectives of this research were to test a previously developed system for verification of the orienting mechanism, and to test the combined performance of four segmentation routines for surface defect detection, using different apple varieties.

The tests were limited to qualitative assessment of the performance of the systems. No attempts were made to address or optimise the performance as far as speed was concerned.

2. Materials and methods

2.1. *Verification of orientation*

In order to avoid problems with apples, not oriented by the orienting system, an additional vision system was designed, with the objective of verifying the orientation. This system comprised a second camera, mounted over the conveyor, at the end of the orienting section, just before the apples entered the tunnel where the image acquisition took place. In this position, the camera was optimally positioned to identify oriented/non-oriented apples. Images of oriented apples, acquired by this camera, will have the apple stem/calyx centred in the image, which facilitates identification. Hence, the presence of a stem/calyx in the centre of the image would mean that the apple was correctly oriented, and could proceed to defect detection. Absence of stem/calyx means that the apple should be recycled to the orienting system.

Images were processed and classified by a method referred to as unsupervised feature extraction. This image-processing method (Bennedsen, 2001) uses the entire image, without any attempt to identify individual features. The method represents an attempt to mimic the way in which humans evaluate and classify objects. The idea is to base the classification on an overall impression of the image as opposed to classical image processing which normally involves identification of the objects by segmentation, followed by the assessment of features like size, shape, number, colour, position, *etc.* In unsupervised feature extraction, no attempts are made to identify features or their value. Groups of images representing different classes are gathered in a matrix, where individual images can be perceived as samples with their pixel values being variables. Subsequently, principal component analysis (Esbernsen *et al.*, 1994) can subject this data matrix to data reduction, and the most significant principal components can be used to train a neural network to perform classification. Unsupervised feature extraction is particularly well suited for situations where a classification is difficult to describe by a number of features and their values. Further, the method can be implemented in such a way that the user can update it without modifying software, simply by providing the system with a number of

examples of the different classes, which is then used to train the neural network.

All image processing, principal component analysis, and data processing were done using Matlab version 6·5·1 with image processing and neural network toolboxes (The Math Works, Inc. Natick, MA, USA).

Initial tests of the concept (Bennedsen *et al.*, 2004) indicated that verification of orienting in this way was indeed possible. Tests under more realistic conditions were carried out using a charge-coupled device (CCD) camera (Sony DFW-X700, Sony Corporation, Tokyo, Japan) mounted over the conveyor, at the end of the orientation section. Two sets of images were acquired: one set of Red Delicious apples, and the other of Golden Delicious. During the test, apples were manually put into the cups of the conveyor at the end of the sorting section in order to obtain a high amount of non-oriented apples. Images of the apples were sorted in groups representing oriented and non-oriented apples. The most obvious examples were used for training the neural networks, while the rest were reserved for test sets. The number of images in the different classes is provided in Table 1.

Prior to performing unsupervised feature extraction, an area around the centre of the image was extracted in order to eliminate the background. Further, equalisation was performed, followed by filtering (*Fig. 1*). The filter was based on a contrast-enhancing filter; it consisted of a 5 by 5 matrix and would enhance the contrast and reduce small variations in grey level.

The images were then reduced to 20 by 20 pixels, vectorised and combined to form a matrix, from which principal components were derived. Based on the number of principal components, neural networks were constructed and trained, using principal components from the training sets. Based on previous experiences with this type of classification, among others (Bennedsen, 2001), three-layer networks were used. The first layer, or input layer, was designed with the number of neurons matching the number of principal components. The second layer, or hidden layer, was the size of, or slightly more than half the size of, the first layer. The output, and final layer, consisted of one neuron. Transfer functions were sigmoid for the two first layers and linear for the output layer. Training was done using back propagation with a Fletcher–Reeves update conjugate gradient algorithm. After training, the performance of the networks was assessed using principal components derived from the test sets.

Table 1
Number of images used for training and tests sets, respectively

Variety	*Class*	*Apples images*	*Training set*	*Test set*
Red Delicious	Non-oriented	111	80	31
	Oriented	401	320	81
	Total	512	400	112
Golden Delicious	Non-oriented	146	116	30
	Oriented	365	319	46
	Total	511	435	76

2.2. *Surface defect identification*

The imaging system on the prototype apple sorter (Throop *et al.*, 1999, 2003a) consisted of a camera (Dalsa 1M60, Waterloo, Ontario, Canada) with high spatial resolution (1024 by 1024 pixels) at high data transfer speeds (40 MHz).

Mounted in front of the camera was an optical splitter that enabled the camera to capture up to three images simultaneously. Band-pass filters in front of the splitter limited the images to consist of selected wavelengths. Based on previous research (Throop *et al.*, 1999), two of the three images were used with filters at 740 and 950 nm, respectively.

While passing the camera, apples were rotated through 360° while the camera acquired six consecutive images. Depending on the system settings and the size of the apple, the frame would show most of the apple

Fig. 1. Image pre-processing: left, original image; centre, stretched and filtered image; right, image reduced to 20 by 20 pixels

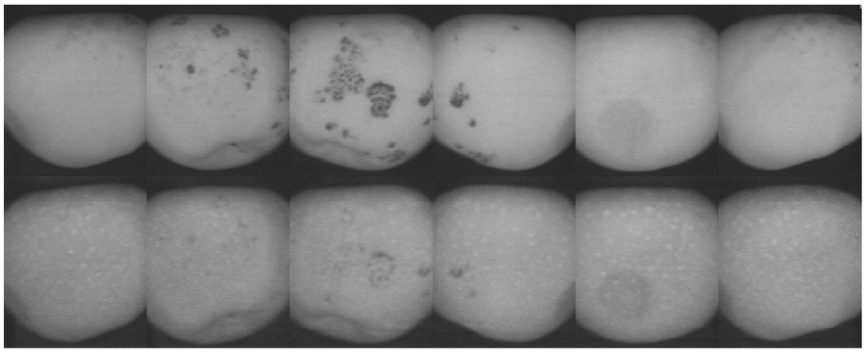

Fig. 2. Original images acquired with the two-waveband filters; top six images are captured through a 740 nm filter, the six images in the lower row through a 950 nm filter

Table 2
Apple varieties used in the test and the dates for picking and testing

Variety	*Picked*	*Date run*	*Time stored*
Gala	End of August	04-02-2004	7 months
Empire	Mid September	13-01-2004	4 months
Jonagold	Mid September	01-04-2004	6·5 months
Golden Delicious	Mid to late September	08-01-2004	3·5 months
Pink Lady	Mid to late September	07-01-2004	3·5 months
Red Delicious	Mid to late September	04-02-2004	4·5 months
Fuji	Mid October	04-03-2004	4·5 months
Rome	Mid to late October	30-10-2003	>1 month

surface facing the camera. An example of images is shown in *Fig. 2*. The upper six sub-images in *Fig. 2* were captured through the 740 nm filter, the lower six images were captured through the 950 nm filter. Typical image frame size was 138 pixels high, in the stem–calyx axis, and 150 pixels wide, corresponding to six frames totalling 138 by 900 pixels per apple image. The frame size was decided during image acquisition and depended on the size of the apples. For practical implementation, only 60° of each apple frame was needed. Specially designed software extracted 60° from the centre of each frame and combined the resulting six frames to an image representing 360°, or the entire surface of the apple. This reduced the final size of an image to 138 by between 350 and 400 pixels, depending on the size of the apple.

The two parts of *Fig. 2* show how the dark spots from various diseases are more evident in the 740 nm range, while the bruises appear darker at 950 nm. The task for the image-processing software was to identify the defects, while not including darker areas from shadows, stem and calyx or the rim of the fruit. Four segmentation methods were implemented to identify the defects. These consisted of three routines based on threshold segmentation, and one based on principal components and neural networks.

2.2.1. *Apple test sets*

Apples of eight varieties were used to test the combined performance of the segmentation routines: Gala, Empire, Jonagold, Golden Delicious, Pink Lady, Red Delicious, Fuji and Rome. With the exception of the Fuji, which was obtained from a commercial grower and packaging enterprise, apples were picked at the Appalachian Fruit Research Station, West Virginia, in the autumn of 2003. Specimens with different surface defects were preferred. Between picking and testing, apples were stored at 0 °C (Table 2). The day before running the apples through the sorting system, bruises were inflicted on most of the apples. This was done by dropping apples 150–200 mm onto the convex surface of a semi-sphere of wood. This created a bruise of about 12–15 mm diameter.

Three classes of defects were identified based on visual inspection of the images: dark marks, bruises and faint marks. Dark marks were generally caused by fungi or bacterial diseases, hail damage or similar, and would normally show up in the 740 nm images. Faint marks were minor defects, which would not cause any downgrading of the apple. The main reason for including faint marks was to avoid counting them as false positives, in case the routines detected them. False

positives are non-defective areas segmented as defective. For each variety, a set of test images was created. It consisted of a total of 200 defects representing all three classes. After selecting the images, the defects were marked. This process was partly manual, and involved marking the defective areas by using the mouse cursor. A routine was developed which calculated the position and the size of each defect based on the marking.

2.2.2. *Thresholding segmentation*

Three different threshold segmentation methods previously developed and tested on Golden Delicious apples were used in this study. They are referred to as simple thresholding, multi-thresholding and intelligent segmentation.

The simple thresholding segmentation was based on routines developed by Throop *et al*. (2003a). Prior to the segmentation, a flat field correction was applied to the images. Flat field correction compensates for the uneven distribution of light, caused by the spherical shape of the apples. In this case, an image of a white sphere, the size of the apples, was inverted and added to the original apple image. However, the flat field correction did not completely eliminate the background and left dark edges around the perimeter of the apples. These edges would be perceived as defects by segmentation using simple thresholding with a single user-selected threshold level. When the threshold was set for a reliable segmentation of the defects, the routine would get the edges as well. *Figure 3* shows an example of a flat field-corrected 950 nm image and the result of segmentation.

In order to prevent the dark background and edges from being identified as defective material, the dark background was removed using a specially designed routine. First, the images were converted into binary images (black and white only). This was done using Matlab's automatic thresholding function, which determines the threshold value based on a histogram of the grey values in the image. The thresholded image showed the position of the apple in the image as a white area on a black background. This black and white image was used as a template to remove the background in the original image. Next, the columns were resized to a standard size of 100 pixels, thus stretching it to fill the entire frame. The coordinates of the border of the apple were recorded so that the results of the segmentation could be resized back to the original size and shape of the apples, and hence allow comparison of the area identified as defective by the segmentation with the actual defects marked on the test set.

The next step was to reduce each of the six frames to 60°. This was done by calculating the perimeter of the apple in the original images. Based on this, a portion representing 60° was extracted from the centre of each of the six frames. Combining these sections yielded an image like the one presented in *Fig. 4*, representing 360°, or the entire surface of the apple. After the extraction of 60° segments, a final trimming of the upper and lower rim of the apple was applied, in order to remove any shadows left by the background removal.

The actual segmentation was done by using Matlab's automatic threshold level function, which would select a level based on a histogram of the grey levels in the image. In case there were no defects in the image, the

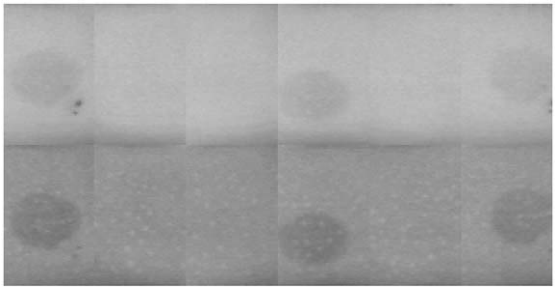

Fig. 4. Images after removal of the background, stretching to a standard frame size of 100 pixels, and extracting six times 60° from the original six frames: top, 740 nm image; bottom, 950 nm image

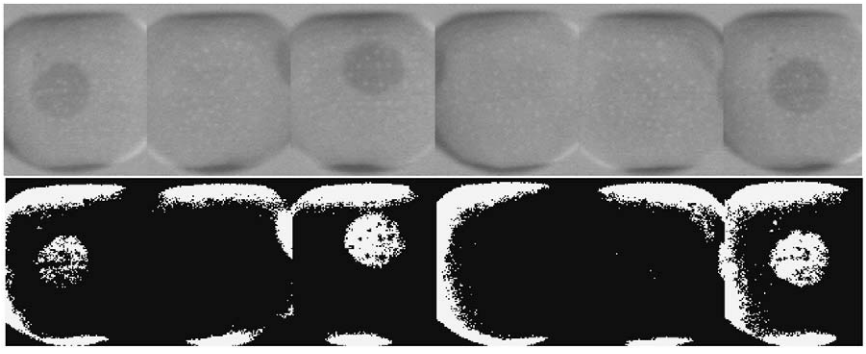

Fig. 3. Top, 950 nm image after flat field correction using a white sphere; bottom, the same image after threshold segmentation

automatic threshold level function would select a level that made local darker areas on the apple surface appear as defects. In order to avoid this, a bar 5 pixels tall was added to the top and bottom of each image. The grey level of this bar was adjusted to appropriate values for different apple varieties and differed for the 740 and 950 nm images. Thus, in the absence of defects, the bar would dictate the threshold level identifying the bar as a defect, instead of local, dark areas. After thresholding, the resulting binary images were resized to their original configuration using the arrays containing the saved coordinate values. Finally, the 'noise' was removed by median filtering, in addition to three erosions followed by two dilations. The final result is shown in *Fig. 5*.

A simple thresholding is not always able to distinguish between defects and darker areas caused by shadows. Further, in many cases a simple thresholding will not identify all of the defective area, due to variations in the grey level within the defective area. The left frame of *Fig. 3* provides a good example. The automatic thresholding routine referred to as 'multi-thresholding' was constructed in order to eliminate these problems. The idea was to construct a binary image that only identified the dark areas representative of defects and not those areas representing shadows. Even though this image would not show the whole area of each defect, it could be used to mark the position of the defects in the original image. With the position of the defects identified, a simple thresholding routine, *e.g.* a gradient segmentation could be employed to determine the area of these defects. In this case, a Matlab library routine was used.

The routine operated on one frame at a time. It located the pixel with the lowest value and used this as a threshold level for the creation of a binary image where the dark areas received the value 'one' and the lighter areas the value 'zero'. Then the threshold value was increased by a user-defined amount and the original image segmented again. The resulting binary image was then added to the first segmented image. This process was repeated until the threshold value reached the maximum pixel value in the original image. The so-called multi-layer image, which was constructed as the sum of the segmented, binary images as described, was a grey-level image in which the areas with the highest pixel values represented the darkest areas in the original image. This image was a combination of a number of layers, corresponding to the number of binary images used to construct it, which in turn equalled the number of times the original image was thresholded.

Fig. 5. Segmented images: top, 740 nm image; bottom, 950 nm image

The multi-layer image was then subjected to segmentation in which the threshold was defined as the maximum value in the multi-layer images minus a user-defined amount. This segmentation aimed at identifying the darkest areas in the original image and the resulting, binary image was referred to as a marker image. The final step consisted in constructing a binary image, based on the marker image and the multi-layer image. With the position of the defects identified, a simple thresholding routine, *e.g.* a gradient segmentation could be employed to determine the area of these defects. In this case, a routine from the Matlab image-processing toolbox was used. The principle is illustrated in *Fig. 6*. On the upper part of the figure, the curve represents pixel values in a line through the multi-layer image. After thresholding and reconstruction, the lower part of *Fig. 6* was created, representing the total area of the defect, but eliminating the shadow area.

This routine referred to as 'intelligent segmentation' was developed as an attempt to improve the performance of bruise identification in the 950 nm image. The routine used the images directly from the camera, with no flat field correction. Instead, a correction image was constructed through a series of median filtering and averages of the six frames of an image. The routine was applied to each individual frame, but relied on all six frames in an image to create the correction image. The routine took the image through a number of steps to the final segmentation, as explained below. First, each image was broken down into its individual six frames (image 1 in *Fig. 7*). Vertical bars, set at a grey level of 70, were added to each side in order to influence the automatic thresholding, similar to the simple segmentation routine. The frames were then filtered with a 10 by 10 median filter (result in image 2 in *Fig. 7*), and the result segmented using Matlab's automatic thresholding function (image 3 in *Fig. 7*).

The filtered image (image 2 in *Fig. 7*) was trimmed down to the size specified by the black and white template (image 3 in *Fig. 7*) and stretched to fit the frame size, yielding image 4 in *Fig. 7*. This image in turn was

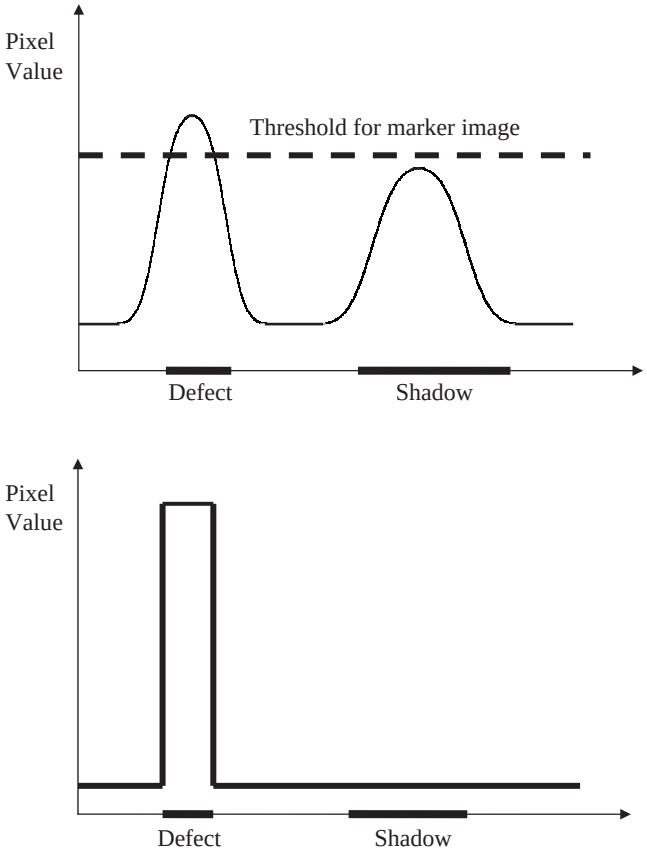

Fig. 6. Graphical representation of the multi-segmentation: in the upper diagram, the curve represents the pixel values in a line through the multi-layer image, constructed based on multiple segmentations of the original image; this image is thresholded, yielding a so-called marker image, which in turn is used to construct the binary image in the lower part, representing the defect, but eliminating the shadow

segmented using Matlab's automatic function (image 5 in *Fig. 7*).

The results of the trimming and stretching (image 4 in *Fig. 7*) for all six frames of the image were added together and divided by 6 in order to construct an average of the six frames (image 6 in *Fig. 7*). In order to ensure that edge defects were not eliminated in the individual frames when the dark edges were removed, an average was made, for each individual frame, of the second segmentation (image 5 in *Fig. 7*) with the average of all segmented frames (image 7 in *Fig. 7*). This new template was used to further trim the individual frames. The six frames, created as results of this trimming, were then averaged and median filtered with a 5 by 5 matrix (image 8 in *Fig. 7*). This image was then inversed in order to create the correcting image which would be added to each frame (image 9 in *Fig. 7*, a visualisation is provided in image 10 by multiplying by 2). The image showed low values corresponding to the highlights in the images to be corrected, but higher values where shadows needed to be lightened.

When the correction image (image 9 in *Fig. 7*) was added to a frame, a uniformly illuminated image was obtained as seen in image 11 in *Fig. 7*. When this image was segmented using the automatic thresholding routine, it yielded the result seen in image 12 in *Fig. 7*. This result for each frame was then resized according to the coordinates saved by an array as it progressed through the trimming functions, and then reduced to 60° of apple surface. In addition, the black and white colours were reversed, the white representing defective material. The final step consisted in stitching the six frames together to show the full apple: the result is shown in *Fig. 8* with the original image inserted for comparison.

2.2.3. *Neural network defect identification*

Neural network identification of defects was based on principal component analysis (PCA) and neural networks in the same way as described for the verification of orientation. The images were perceived as data sets, in which each individual column was considered a sample, and the pixel values as variables. These data sets were then subjected to PCA in order to reduce the number of variables. By using PCA, the columns of pixels are substituted by a column of principal components (PCs). The advantage of this method is that the number of PCs was considerably lower than the number of pixels, and that the PCs provided an optimised basis for classification. Each column was assigned a value of 'zero' if it did not represent a defect, and 'one' if it did represent a defect. The new matrix, consisting of columns of PCs, and a corresponding target vector of 'zeros' and 'ones' were used to train the neural networks. As described for the orientation verification, three-layer networks were used. The first layer, or input layer, had a number of neurons matching the number of principal components. The second layer, or hidden layer, contained slightly more than half the size of the first layer. The output, and final layer, consisted of one neuron. Transfer functions were sigmoid for the two first layers and linear for the output layer. Training was done using back propagation with a Fletcher–Reeves update conjugate gradient algorithm.

Due to the difference in the detecting ability of the two wavebands used, two training sets were constructed for each variety: one for 740 nm images, and one for 950 nm. Images for the training sets were chosen by the presence of clear and typical defects. After selecting the

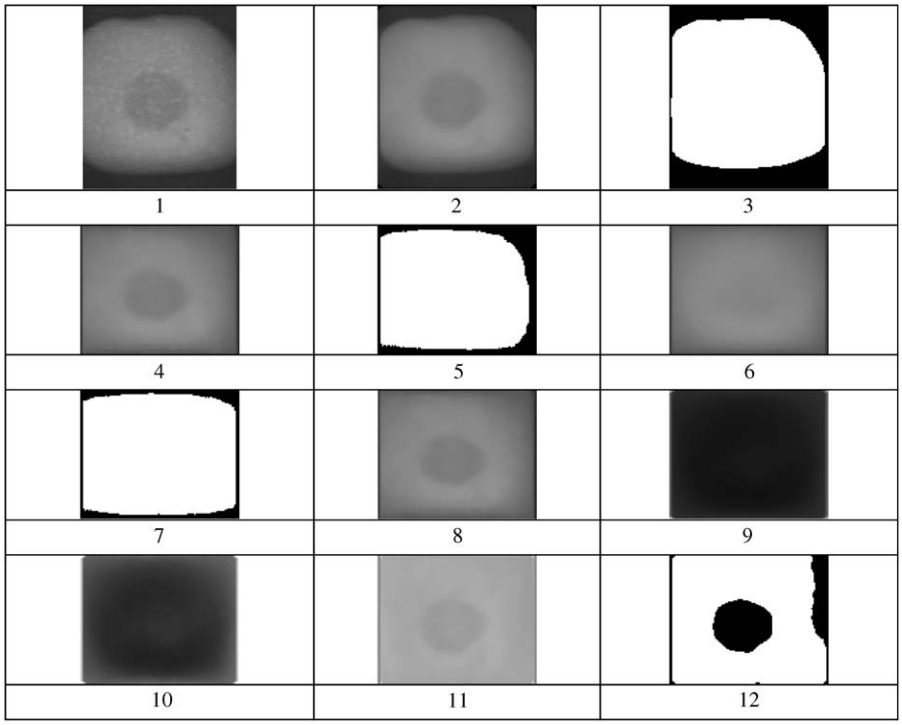

Fig. 7. 'Intelligent segmentation': 1, original first frame; 2, influencing bars added and filtered; 3, result of segmentation; 4, filtered image after trimming and stretching; 5 same image after segmentation; 6, average of six filtered and trimmed images; 7, average the same six images after segmentation; 8, median filtered average of six trimmed frames; 9, inverse of 8; 10, visualisation of 9 by multiplying by 2; 11, individual frame after application of the correction image (9 Fig. 6); 12; Same image after segmentation

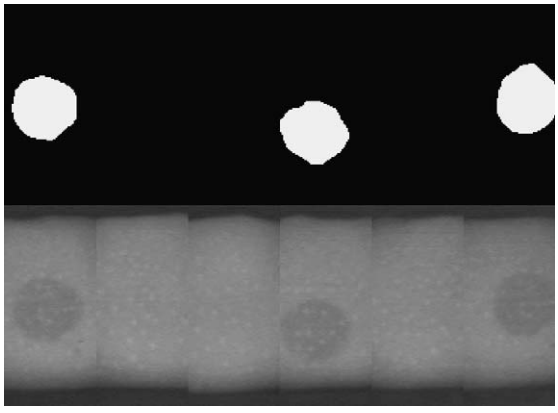

Fig. 8. Result of the 'intelligent segmentation' (top) and original image (bottom)

images, the defects were marked as mentioned for the test sets. Based on the manual marking, a routine performed the image processing which created a target vector of 'zeros' and 'ones', depending on whether the area was marked as defective or not. The neural networks at 740 nm were marked and trained to detect primarily dark spots from various diseases, while the neural networks based on 950 nm images were constructed and trained in a similar way to detect the bruises. The routine also allowed certain areas of the training image to be eliminated, such as artefacts, shadows, and stem or calyx, which may confuse the neural network and impede training. The routine then created a new image without the unwanted areas. The resulting training set images and their corresponding target vectors were merged to form an image containing 12 to13 apple images and consisted of between 5000 and 6000 samples (columns of pixels) of which about 30–40% represented defects, and the rest non-defective surface.

After training, the networks were tested using the test sets containing apple images, new to the system. Principal components were derived using the eigenvectors generated in connection with the training set. An example of the output of a network is shown in *Fig. 9*, with the original image inserted above the graph.

Different versions of the neural network approaches were tested. These included organising the images such that the pixel columns were vertical, *i.e.* parallel to the stem–calyx axis, and horizontal, *i.e.* perpendicular to the stem–calyx axis, and combining the two. These tests

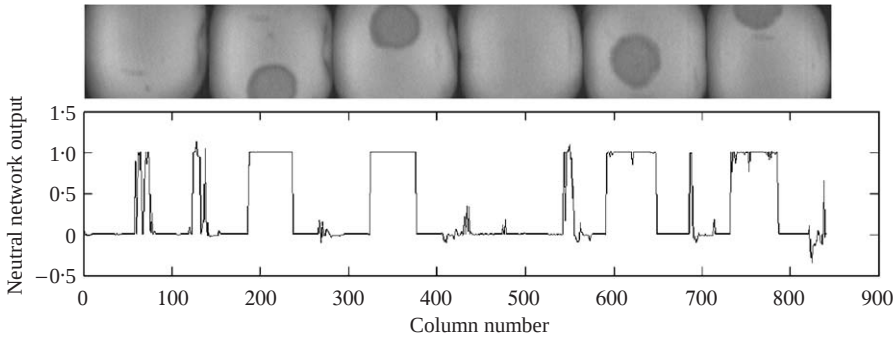

Fig. 9. Example of neural network detection of defects; the image containing six sub-frames is lined up with a plot of the network test output; the network assigned the value zero to columns not found to include a defect, and one to columns including a defect

showed that the best results were obtained with samples parallel to the stem–calyx axis.

2.2.4. *Combined defect identification*

During the development and testing of the segmentation routines, it was found that their performance depended on the type of defects and the image (740 or 950 nm). Tests using Golden Delicious apples showed that multi-thresholding based on the 740 nm images was most efficient for detection of large and small dark marks. The faint dark marks were best detected using simple thresholding with the 740 nm images and multi-thresholding with the 950 nm images. Intelligent segmentation proved to be the best way of detecting bruises in the 950 nm images. Dark marks and bruises at the edge of the apples could best be detected using multi-thresholding with the 950 nm images for edge bruises and 740 nm for dark marks. The different routines would normally not indicate false positives at the same positions. This means that by combining the results of the routines, one could confirm the defects identified by another, while eliminating false positives.

A number of different ways of combining the output of the four segmentation methods were considered. However, in the end it was realised that a simple 'voting' by the routines was adequate. The way in which it was done was by adding the four resulting images, and then dividing the result by 2. In this way, if two or more routines identified a defect, it was accepted. If only one routine suggested a defect, it would be discarded. As the defects were marked by white, or 255 on a grey scale, two or more routines agreeing on the presence of a defect would yield a final value of 255 when the result was converted to 8-bit integer. If only one routine indicated the presence of a defect, the value would be 127·5, which was converted to zero.

3. Results and discussion

3.1. *Orientation*

For the Red Delicious images, nine principal components were derived, for an accuracy of 0·02. This means that the ninth principal component is the last which contributes more than 2% to the classification. A neural network was constructed, consisting of nine input neurons, five neurons in the hidden layer and one output neuron. Trained with the training set, and using back propagation and the Fletcher–Reeves update conjugate gradient algorithm, the network reached minimum gradient in 1001 training epochs, yielding a mean square error of $1{\cdot}311 \times 10^{-5}$. (The set target was 1×10^{-5} for mean square error).

For the golden Delicious images, 13 principal components were derived, for an accuracy of 0·02. The neural network consisted of 13 input neurons, seven neurons in the hidden layer and one output neuron. Trained with the training set, and using back propagation and the Fletcher–Reeves update conjugate gradient algorithm, the network converged to the set goal of 1×10^{-5} for mean square error in 527 training epochs.

Results of testing the neural networks on the test sets (Table 1) are shown in Tables 3 and 4. In a practical implementation, the oriented apples, which are misclassified as non-oriented, do not represent a problem in relation to defect detection since these apples would be recycled to the orienting system. This would however reduce the capacity of the system. Problems occur when non-oriented apples are not identified. For these apples, stem or calyx may be perceived as defects by the defect-sorting system, and the apples will be rejected. The tests indicate that this happens in 3% of the cases for Golden Delicious (Table 4) and in 16% of the cases for Red Delicious (Table 3). In order to evaluate the significance of these numbers, it should be taken into consideration

Table 3
Results of classifying test set of Red Delicious images

	Classification		*Error, %*
Class	*Non-oriented*	*Oriented*	
Non-oriented	26	5	16
Oriented	10	71	12

Table 4
Results of classifying test set of Golden Delicious images

	Classification		*Error, %*
Class	*Non-oriented*	*Oriented*	
Non-oriented	29	1	3
Oriented	8	38	17

that test sets consisted of images which did not show a clear stem or calyx in the centre of the images, or the absence for the non-oriented apples; instead, the test set consisted of images where there could be some doubt as to whether or not the apple was actually oriented. For the Red Delicious in particular, their somewhat irregular shape made it even more difficult to determine if the apples were in fact oriented. Indeed, a closer inspection of the five images apparently wrongly classified as oriented revealed that only one image was definitely not oriented, and that the rest of the apples had the stem–calyx axis angled in relation to the vertical, probably still passing the defect sorter in a correct attitude. This brings the error down to 3%.

In a practical application of the orienting system, a maximum of 5% of the apples will not be oriented. With an error rate corresponding to 3 to 4% of those not identified by the system, the potential loss becomes 0·15–0·2% of all the apples passing the sorter. This number is slightly higher than the one found during the laboratory testing of the concept. In this case, an overall error of 0·05% was indicated (Bennedsen & Peterson, 2004) however, this test used a larger training set and the images were reduced to 24 by 24 pixels as opposed to 20 by 20 pixels in this study.

3.2. *Defect detection*

The results of processing the images acquired through the 740 and the 950 nm filters individually are shown in *Figs 10–13*. The data are presented in tables in the appendix. Dark marks are easier to detect in the 740 nm images and some of them do not appear in the 950 nm images. Bruises will normally be shown in both images, however, with considerably more contrast in the 950 nm images. The figures show number and area of defects detected for each class, and the number of false positives. False positives are the number of pixels misclassified as defects, in per cent of the total number

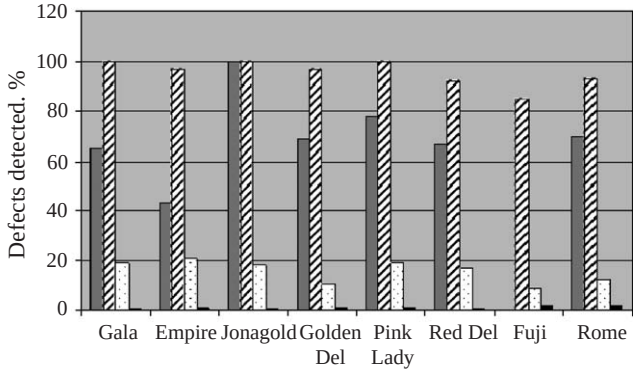

Fig. 10. Results of defect detection using the 740 nm filter: ■, dark marks; ▨, bruises; ⬚, faint marks; ■, false positives

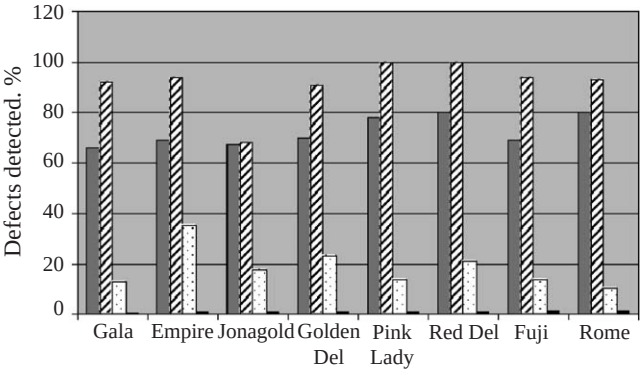

Fig. 11. Results of defect detection using the 950 nm filter: ■, dark marks; ▨, bruises; ⬚, faint marks; ■, false positives

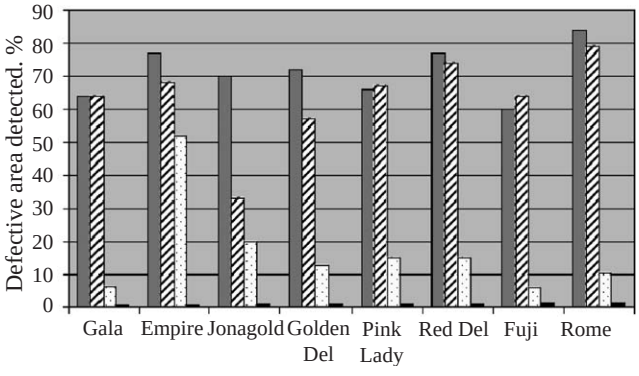

Fig. 12. Results of defect detection using the 740 nm filter. Area of defects: ■, dark marks; ▨, bruises; ⬚, faint marks; ■, false positives

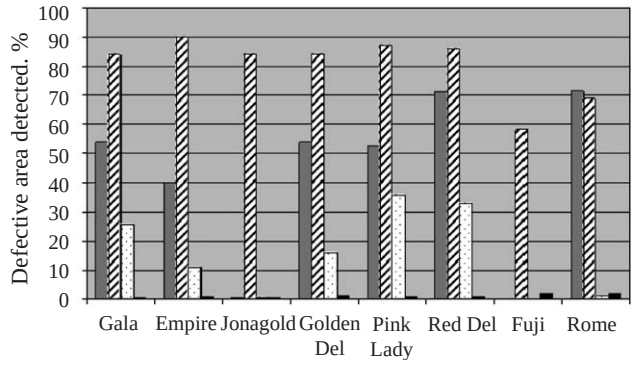

Fig. 13. Results of defect detection using the 950 nm filter. Area of defects: ■, dark marks; ▨, bruises; ⬚, faint marks; ■, false positives

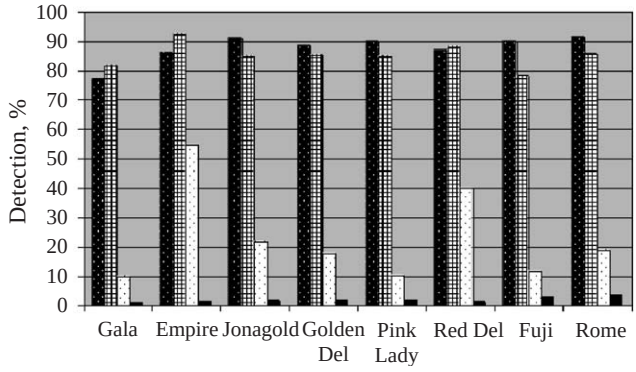

Fig. 14. Combined results of defect detection using a combination of the 740 and 950 nm images: ■, number; ▨, area; ⬚, faint marks; ■, false positives

of pixels in the image. A defect was considered detected if 10% of its area was identified by the combined segmentation procedure.

The Fuji apples were provided by a commercial grower who sorted the apples before supplying them. Hence, only very few dark marks were found on this variety, and none of them showed up in the 950 nm images. Further the Fuji apples were waxed, which may have impeded the detection.

In order to evaluate the overall performance of the system, a combination was made of the resulting, binary images from the combined segmentations of the 740 and 950 nm images. Similarly, a test set was made by combining the 740 and 950 nm images in which the defects were marked. During the merging of the test sets, care was taken to ensure that redundant markings were removed and that the size of the defects were chosen according to the best representation, *i.e.* 740 nm images for dark marks and 950 nm images for bruises. The combined result of the segmentation was then compared with the combined test set. The results for the number of defects and area are presented in *Fig. 14*. The faint marks were left out of both the total number of defects and the number of detected defects. The rationale behind this was that faint marks were not considered the cause for declassification of the apples; however, when detected they should not count as false positives.

In a practical implementation, the settings for the thresholding routines will have to be adjusted for the different varieties. Further, it should be considered to train a neural network for each variety. Tests showed that neural networks, trained on *e.g.* Golden Delicious will perform reasonably on *e.g.* Rome and Pink Lady apples; however better results were obtained with networks trained for the actual variety. It will be necessary to make allowance for the false positives. An average level of 2% was obtained, but given the differences between varieties, it will be better to use individual values. This means that only apples with more than a threshold number of pixels, identified as defective, should be considered as possessing surface defects. With a 2% level of false positives in a 138 by 375 pixels image, this amounts to 1035 pixels. Further allowance will have to be made for the faint marks in whether or not these are to be included as defects. This will depend on the actual quality criteria.

4. Conclusion

A system for identifying surface defects on apples was established and tested using eight different varieties of apples. The orientation-verification system successfully identified non-oriented apples with an error rate for practical implementation of 0·15–0·2%. The defect detection system identified between 77 and 91·6% of the defects, with 1·1–3·6% false positives. Combined area of defects identified ranged from 78 to 92·7%. The industry will ask for 100% detection with no false positives. However, this is not obtained with the current, manual sorting albeit the level of detection is not known. Whether or not the performance of the system is adequate depends on existing quality criteria.

References

Aneshansley D J; Throop J A; Anger W C; and Peterson D L (2003). A multivision linear filter for capturing multispectral images. ASAE Paper No. 03-3027

Bennedsen B S (2001). Use of unsupervised feature extraction for sorting fruits and vegetables. In: 'Fruit, Nut, and Vegetable Production Engineering', Proceedings of the 6th

International Symposium, Potsdam, pp 357–361. ATB, Institut für Agrartechnik, Bornim

Bennedsen B S; Peterson D L (2004). Identification of apple stem and calyx using unsupervised feature extraction. Transactions of the ASAE, **47**(3), 889–894

Crowe T G; Delwiche M J (1996). Real-time defect detection in fruit—Part I and II: design concepts and development of prototype hardware. Transactions of the ASAE, **39**(6), 2299–2317

Esbernsen K; Schönkopf S; Midtgaard T (1994). Multivariate Analysis in Practice. Camo ASA, Oslo, Norway. ISBN 82-993330-1-6

Leemans V; Destain M F; Magein H; Herregods M (ed.); Boxus P (ed.); Baets W (ed.); Jager A (2000). Quality fruit grading by colour machine vision: defect recognition. Proceedings of the XXV International Horticultural Congress. Part 7. Quality of Horticultural Products: Starting Material, Auxiliary Products, Quality Control, Brussels, Belgium, 2–7 August, 1998. Acta-Horticulturae, **517,** 405–412

Throop J A; D J Aneshansley (1997). Apple damage segmentation utilizing reflectance spectra of the defect. ASAE Paper No. 97-3078

Throop J A; Aneshansley D J; Anger W C (1999). Inspection station detects defects on apples in real time. ASAE Paper No. 99-3205.

Throop J A; Aneshansley D J; Anger W C (2000). A multispectral inspection station detects defects on apples. Photonics East 2000, Conference 4203-Biological Quality and Precision Agriculture II, November 7–8, 2000, Boston, MA

Throop J A; Aneshansley D J; Anger W C; Peterson D L (2003a). Quality evaluation of apples based on surface defects—an inspection station design. ASAE Paper No. 03-6161

Throop J A; Aneshansley D J; Anger W C; Peterson D L (2003b). Conveyor design for apple orientation. ASAE Paper No. 03-6123

Appendix AResults of the defection for the eight different apple varieties (Tables A1–A5).

Table A1
Combined results of defect detection based the combination of 740 and 950 nm images

Variety	*Defect detection, %*			*False positives, %*
	Number	*Area*	*Faint marks*	
Gala	77	82	10	1·1
Empire	86·25	92·7	55	1·54
Jonagold	91	85	22	2
Golden Del.	88·5	85·5	17·6	2
Pink Lady	90	85	10·4	1·84
Red Del.	87	88	40	1·5
Fuji	90	78	12	3·07
Rome	91·6	86·1	19	3·6

Table A2
Results of defect detection based on 740 nm images. Number of defects identified

Variety	*Defect detection, %*			*False positives, %*
	Dark marks	*Bruises*	*Faint marks*	
Gala	66	92	13	0·64
Empire	69	93·75	35	0·8
Jonagold	67	68	18	0·98
Golden Del.	70	90·6	23	1·1
Pink Lady	78	100	14	0·86
Red Del.	80	100	21	1
Fuji	69	94	14	1·3
Rome	80	93	10·4	1·5

Table A3
Results of defect detection based on 950 nm images. Number of defects identified

Variety	*Defect detection, %*			*False positives, %*
	Dark marks	*Bruises*	*Faint marks*	
Gala	65	100	19	0·47
Empire	43	97	21	0·69
Jonagold	100	100	18	0·53
Golden Del.	69	97	10	1·1
Pink Lady	78	100	19	0·8
Red Del.	67	92	17	0·62
Fuji	*	85	9	1·9
Rome	70	93	12	1·8

* No data.

Table A4
Results of defect detection based on 740 nm images. Defective area identified

Variety	*Defect detection, %*			*False positives, %*
	Dark marks	*Bruises*	*Faint marks*	
Gala	64	64	6·4	0·64
Empire	77	68	52	0·8
Jonagold	70	33	20	0·98
Golden Del.	72	57	12·6	1·1
Pink Lady	66	67	15	0·86
Red Del.	77	74	15	1
Fuji	60	64	6·1	1·3
Rome	84	79	10·4	1·5

Table A5
Results of defect detection based on 950 nm images. Defective area identified

Variety	*Defect detection, %*			*False positives, %*
	Dark marks	*Bruises*	*Faint marks*	
Gala	54	84	26	0·47
Empire	40	90	11	0·69
Jonagold	0·5	84	0·3	0·53
Golden Del.	54	84	16	1·1
Pink Lady	52·5	87	36	0·8
Red Del.	71	86	33	0·62
Fuji	*	58	*	1·9
Rome	71·6	69	1	1·8

* No data.